

中華民國經濟部智慧財產局
INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA

JCS64 U.S. PTO
09/767191
01/22/01

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，
其申請資料如下：

This is to certify that annexed is a true copy from the records of this
office of the application as originally filed which is identified hereunder:

申請日：西元 2000 年 07 月 03 日
Application Date

申請案號：089113261
Application No.

申請人：行政院國家科學委員會
Applicant(s)

局長
Director General

CERTIFIED COPY OF
PRIORITY DOCUMENT

陳明邦

發文日期：西元 2000 年 10 月 27 日
Issue Date

發文字號：08911015421
Serial No.

申請日期： 89 7 3	案號： 89113261
類別：	

(以上各欄由本局填註)

發明專利說明書

一、 發明名稱	中 文	多段編碼結構對應至單一表格的索引法
	英 文	An Indexing Method for Mapping Multiple Segments of Coded Fields into a Table-Structure Field
二、 發明人	姓 名 (中文)	1. 邱孟希 2. 陳景章 3. 張慶龍 4. 侯廷昭
	姓 名 (英文)	1. 2. 3. 4.
	國 籍	1. 中華民國 2. 中華民國 3. 中華民國 4. 中華民國
	住、居所	1. 台南市安南區本原街三段266號 2. 嘉義縣民雄鄉三興村62之1號 3. 雲林縣斗六市鎮東路360號 4. 嘉義縣民雄鄉東榮村中庄109號5F之6
三、 申請人	姓 名 (名稱) (中文)	1. 行政院國家科學委員會
	姓 名 (名稱) (英文)	1.
	國 籍	1. 中華民國
	住、居所 (事務所)	1. 台北市和平東路2段106號18樓
	代表人 姓 名 (中文)	1. 翁政義
	代表人 姓 名 (英文)	1.



申請日期：	案號：
類別：	

(以上各欄由本局填註)

發明專利說明書

一、 發明名稱	中 文	
	英 文	
二、 發明人	姓 名 (中文)	5. 龔育正
	姓 名 (英文)	5.
	國 籍	5. 中華民國
	住、居所	5. 雲林縣斗六市合作街120巷9號
三、 申請人	姓 名 (名稱) (中文)	
	姓 名 (名稱) (英文)	
	國 籍	
	住、居所 (事務所)	
	代表人 姓 名 (中文)	
	代表人 姓 名 (英文)	



四、中文發明摘要 (發明之名稱：多段編碼結構對應至單一表格的索引法)

本發明是一種將多段的編碼結構對應到單一表格的索引法，這種索引法不但在很短的延遲內就可運作完成而且只要所有編碼結構變動的總合位元數等於索引表格的位元寬度就可達到無碰撞的特性。這個索引法可以利用來做非同步傳輸模式中 VPI/VCI 的對應方法。

英文發明摘要 (發明之名稱：An Indexing Method for Mapping Multiple Segments of Coded Fields into a Table-Structure Field)

This invention proposes an indexing method for mapping multiple segments of coded fields into a table-structure field. This indexing method can complete resolving address mapping within a short delay, and it has a collision-free characteristic when total bit-width of variation for coded fields is the same as the bit-width of the table. This method can be used to resolve VPI/VCI mapping for the ATM technologies.



本案已向

國(地區)申請專利

申請日期

案號

主張優先權

無

有關微生物已寄存於

寄存日期

寄存號碼

無

五、發明說明 (1)

產業上之利用領域

本發明提出應用在非同步傳輸模式(ATM, Asynchronous Transfer Mode)有關處理 VPI/VCI 連線的方法；此方法也可應用在非同步傳輸模式(ATM)交換機，配合共享記憶體(Shared Memory)的觀念在索引表格記憶體的控制上，這樣的架構可在一定大小的共享記憶體的情況下，更有效的調整每一個接收埠支援的連線數。

背景

在非同步傳輸模式網路(ATM, Asynchronous Transfer Mode)中是以 *VPI/VCI* 來區別不同的連線(ATM connection)，一個非同步碼格只有 53 個位元組，再加上日益快速的傳輸速度，使得非同步網路的接收端必須在很短的時間內處理好一個碼格(Cell)的 *VPI/VCI* 的辨認及處理；對網路卡的接收端而言能夠成功地接收到上層，而對交換機的接收端而言能很快找到下一個階段的輸出埠以及新的 *VPI/VCI* 值。

UNI(User-Network Interface)環境下 *VPI* 佔了 8 個位元，*VCI* 佔了 16 個位元，NNI(Network-Network Interface)環境下 *VPI* 佔了 12 個位

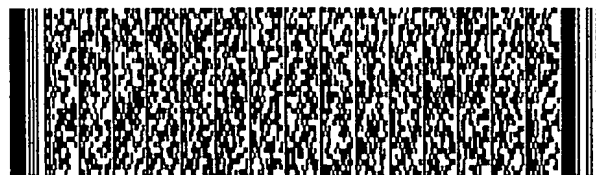


五、發明說明 (2)

元，*VCI* 佔了 16 個位元，因此在 UNI、NNI 中，*VPI/VCI* 總共的位元數分別是 24 位元以及 28 位元，使用與總合位元數相同的位元寬度表格來直接索引的話，雖然可以直接索引到相關的資料，但需要有 2^{24} (UNI) 或 2^{28} (NNI) 個索引表格資料條目 (Entry)，因此硬體成本太高。

因為非同步傳輸模式網路每一個接收端 *VPI/VCI* 的連線數一般不會多到 2^{24} 或 2^{28} 條，因此在 ATM 接收端要分辨每一條不同的 *VPI/VCI* 連線實現的較複雜的方法有兩種，第一種是利用計算機科學中赫序 (Hashing) 方法使用軟體或硬體 (ASIC) 來實現，其軟硬體較複雜，第二種是利用內容定址記憶體 (CAM, Content Addressable Memory)，但內容定址記憶體 (CAM) 在連線數多的情況下，硬體成本很高，這兩種方法由於成本花費較高，因此不在本發明討論的範圍之內。

本發明主要在探討索引函數 (Indexing Function) 的理論，如圖 1 所示為可實現在非同步傳輸模式 (ATM) 接收端的 *VPI/VCI* 索引法的示意圖，在此 *VPI/VCI* 當作輸入，經過索引函數 (Indexing Function) 運算後，產生 n 位元的索引值，因為每個資料條目佔的字組數 (word) 是 W ，



五、發明說明 (3)

因此要將 n 位元索引值先乘以 W 得到對應索引表格的抵補值(offset)值，再將實際存放索引資料的基礎位址(Base Pointer)加上，據以找到相對應的索引資料條目(Entry)。圖 1 所示之索引函數的作法常用的簡單方式有截取位元 (Truncation)法以及互斥或運算 (exclusive-OR)法。

在探討截取位元 (Truncation)法及互斥或運算 (exclusive-OR)法以前，我們知道交換機規範 VPI/VCI 號碼選用有兩種常見的方法，第一種是對非同步傳輸模式(ATM)交換機 VCI 的上、下限作設定， VPI 號碼的選用由使用者依自己需要做設定，第二種方式是設定 VPI 及 VCI 的可變化幾個較低位元(LSB, Least Significant Bit)。

使用截取位元 (Truncation)法的 ATM 網路卡可將 VPI 及 VCI 使用到的位元數作調整，並直接使用這些位元作索引的動作以避免不同的 VPI/VCI 值索引到同一個索引值(我們把不同的 VPI/VCI 值索引到同一個索引值的現象稱為碰撞(collision))，這種網路卡若搭配以上討論的非同步傳輸模式(ATM)交換機，須設定交換機以及網路卡要截取那些位



五、發明說明 (4)

元來做索引。而在本發明提出之反向補償索引法(Reverse Compensation Indexing Method)相對於使用截取位元法的網路卡為只需設定交換機即可，不必再規畫網路卡本身，接下來為了說明一般使用的互斥或運算(exclusive-OR)法的缺點，以下舉一個簡單的實施例來說明。

有一種使用互斥或運算(exclusive-OR)法在索引位元寬度為 0 到 12 位元寬度是以如下的運算式做為索引值：

$$(\{VPI[7:0], VCI[15:12]\} \oplus VCI[11:0]) \& (2^{TAB_SIZE} - 1)$$

其中 \oplus 表示逐位元的互斥或運算， $\&$ 表示是逐位元的及(AND)運算，

TAB_SIZE 為 Table Size 的縮寫，表示索引值的位元寬度。

為了說明這種方式不適用的場合，在此我們舉個實施例：

若 $TAB_SIZE=7$ ，使用互斥或運算(exclusive-OR)索引運算式

$$(\{VPI[7:0], VCI[15:12]\} \oplus VCI[11:0]) \& (2^{TAB_SIZE} - 1)$$

$$=(\{VPI[7:0], VCI[15:12]\} \oplus VCI[11:0]) \& (2^7 - 1)$$

$$=(\{VPI[7:0], VCI[15:12]\} \oplus VCI[11:0]) \& (1111111)_2$$

$$=(\{VPI[2:0], VCI[15:12]\} \oplus VCI[6:0])$$



五、發明說明 (5)

若非同步傳輸模式(ATM)交換機規畫目前的 VPI ， VCI 使用的區間為：

$$\min(VPI) = 0, \max(VPI) = 0, \min(VCI) = 0, \max(VCI) = 127 = (1111111)_2$$

則不會產生任何索引值的碰撞，但若是以下的情形：

$$\min(VPI) = 0, \max(VPI) = 1, \min(VCI) = 0, \max(VCI) = 63 = (111111)_2$$

我們至少可舉出一個會產生碰撞的情形：

$(VPI/VCI)=(0/16)$ 運算式為

$$\{VPI[2:0], VCI[15:12]\} \oplus VCI[6:0]$$

$$=(0000000)_2 \oplus (0010000)_2$$

$$=(0010000)_2$$

另外一對 $(VPI/VCI)=(1/0)$ 運算式為

$$\{VPI[2:0], VCI[15:12]\} \oplus VCI[6:0]$$

$$=(0010000)_2 \oplus (0000000)_2$$

$$=(0010000)_2$$

很明顯的，至少有 $(0/16)$ 及 $(1/0)$ 均索引到同樣的索引值，產生碰撞。

換句話說，沒有本發明之觀念設計出來的互斥或運算 (exclusive-OR) 法，在只有一條 VPI 時搭配以上討論的非同步傳輸模式 (ATM) 交換機最多可達到相等於索引位元寬度索引值無碰撞的效果，而



五、發明說明 (6)

在不只一條 VPI 時，如果 VPI 的號碼是連續的話會造成碰撞，這個情形發生的原因是因為 VPI 的位元零($VPI[0]$)若與 VCI 會變化的位元作互斥或(如上例為 VCI 的位元四($VCI[4]$))，碰撞就會發生。選取適當的 VPI 號碼組可避免碰撞的發生。但相對的設定交換機的工作也隨之增加，這種選取 VPI 號碼組的方式適合於前面提及第一種設定方法的非同步傳輸模式(ATM)交換機。而使用第二種方法的非同步傳輸模式(ATM)交換機是透過設定 VPI 較低幾個位元變動(LSB, Least Significant Bit)，顯然不適合搭配使用互斥或運算(exclusive-OR)法的網路卡來運作。

為了改善以上現行非同步傳輸模式(ATM)網路運作的環境，特別是在避免碰撞上，下一節會先證明本發明中多段編碼結構對應至單一表格的索引法，並據以應用在非同步傳輸模式(ATM)網路中兩段編碼結構 VPI/VCI 的反向補償(Reverse Compensation)索引法。我們會說明這個富使用彈性的反向補償索引法確實改善了目前的作法。以下 max 表示 maximum 是上限、最大值的意思，而 min 表示 minimum 是下限、最小值的意思。反向補償索引法的卓越特性為只要 $\min(VPI)=0$ 及 $\min(VCI)=0$ ，用同一套電路可適用於：



五、發明說明 (7)

$\max(VPI) = 2^{k_0} - 1$
 $\max(VCI) = 2^{n-k_0} - 1$, $k_0 = 0, 1, \dots, n$ 等 $(n+1)$ 種場合都具有索引值無碰撞的效果，如圖 1 所示為可實現在非同步傳輸模式(ATM)接收端的 VPI/VCI 索引法的示意圖，整個索引法可區分成三個階段：索引值運算、位址產生、記憶體存取，吾人令索引值運算時間是 t_i ，位址產生時間是 t_a ，記憶體存取時間是 t_m ，則反向補償索引法的電路運作於整個索引法中可支援到每秒 $1/(t_i + t_a + t_m)$ 個 ATM 碼格(Cell)的速率，若互斥或閘的延遲為 t_x (單位為秒)，則此時之 $t_i = t_x$ 。

發明目標

本發明之首要目的係揭示一種將多段的編碼結構對應到單一表格索引法的概念與實施方法，這種索引法不但在很短的延遲內就可運作完成而且只要所有編碼結構變動的總合位元數等於索引表格的位元寬度就可達到無碰撞的特性。

本發明之次要目的係揭示一種反向補償索引法的實施方法，只要 $\min(VPI)=0$ 及 $\min(VCI)=0$ ，用 n 個互斥或閘的電路可適用於：



五、發明說明 (8)

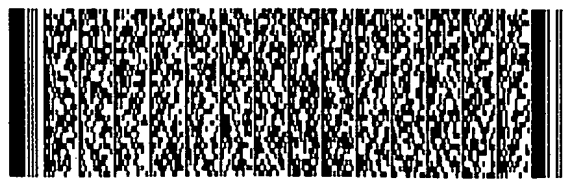
$\max(VPI) = 2^{k_0} - 1$
 $\max(VCI) = 2^{n-k_0} - 1$, $k_0 = 0, 1, \dots, n$ 等 $(n+1)$ 種場合都具有索引值無碰撞的效果。

本發明尚且揭示一種運用多段編碼結構對應到單一表格的方法，以及運用多段編碼結構對應到共享記憶體(Shared Memory)的架構，可在一定大小的共享記憶體的情況中更有效地調整每一個接收埠支援的連線數。

凡是熟悉該技藝的人士在閱讀下列經由不同圖解所展示之較佳實施例詳細說明後，無疑地將非常清楚本發明所揭示之目的和優點。

發明之詳細說明

本發明先針對多段編碼結構對應到單一表格的理論作探討，並據以證明本索引法具備之特性；再對如 VPI/VCI 的兩段編碼結構作探討，由此導引出反向補償索引法(Reverse Compensation Indexing Method)的使用彈性，並舉出本索引法在非同步傳輸模式(ATM)網路設備如網路卡及交換機的接收單元的應用。



五、發明說明 (9)

若有 m 段的編碼結構，此 m 段編碼結構由 \bar{A}_i^m , $i=0,1,\dots,m-1$ 組成， \bar{A}_i^m 最多有 2^{k_i} 種變化。我們可利用某種轉換方法為將原本的 \bar{A}_i^m 各個轉換為 \bar{A}_i^n ，且 \bar{A}_i^n 有 k_i 個位元會變動， $\sum_{i=0}^{m-1} k_i = n$ 。 n 為本索引法的索引位元寬度， k_i 的大小會影響整個總合 n 的大小，也就是索引表格的位元寬度，每一個 k_i 大小取決於轉換技巧的優劣。

前一段落提及將 \bar{A}_i^m 轉換為 \bar{A}_i^n 的方法有很多種，本發明提供兩種可依需要個別或混合使用的方法。這兩個方法的目的均是使得每一 k_i 能夠儘量小，以期減少索引位元度 n ，其中 $\sum_{i=0}^{m-1} k_i = n$ 。第一種轉換的方法是觀察原本的每一個 \bar{A}_i^m ，若會變動的位元並不是在最右側，則可將會變動的位元調整到最右側。

第二種轉換方法可適用於以下的條件下：若其中某些編碼結構的值為 $\min(\bar{A}_i^m) \leq \bar{A}_i^m \leq \max(\bar{A}_i^m)$ ， k_i 滿足以下的數學式：

$$2^{k_i-1} < (\max(\bar{A}_i^m) - \min(\bar{A}_i^m) + 1) \leq 2^{k_i}$$

我們可將這些編碼結構 \bar{A}_i^m 分別加或減某值使轉換後的 \bar{A}_i^n 低 k_i 個位元會變動。可以減的值的特例剛好是 $\min(\bar{A}_i^m)$ 。



五、發明說明 (10)

\bar{A}_i 均跟 $(2^n - 1)$ 做逐位元的 AND 運算，可調整為 n 位元寬度的 \bar{A}_i 。

我們以英文大寫表示是固定值的位元，英文小寫表示是會變動的位元。

將 \bar{A}_i 調整成如下的型式作 $\bar{A}_0 \oplus \bar{A}_1 \oplus \dots \oplus \bar{A}_{m-1}$ 運算； \bar{A}_i 的第 $n - (\sum_{j=0}^{i-1} k_j + 1)$ 到第 $n - \sum_{j=0}^i k_j$ 位元均為會變動的位元。（在此為使表示式簡潔起見，令

$$\sum_{j=0}^{i-1} k_j = 0$$

$$\bar{A}_0 \oplus \bar{A}_1 \oplus \dots \oplus \bar{A}_{m-1} =$$

$$(a_{0_{n-1}}, a_{0_{n-2}}, \dots, a_{0_{n-k_0}}, A_{0_{n-k_0-1}}, A_{0_{n-k_0-2}}, \dots, A_{0_{n-k_0-k_1}}, \dots, A_{0_{n-(\sum_{i=0}^{m-2} k_i)}}, A_{0_{n-(\sum_{i=0}^{m-2} k_i)-1}}, A_{0_{n-(\sum_{i=0}^{m-2} k_i)-2}}, \dots, A_{0_0})$$

\oplus

$$(A_{1_{n-1}}, A_{1_{n-2}}, \dots, A_{1_{n-k_0}}, a_{1_{n-k_0-1}}, a_{1_{n-k_0-2}}, \dots, a_{1_{n-k_0-k_1}}, \dots, A_{1_{n-(\sum_{i=0}^{m-2} k_i)}}, A_{1_{n-(\sum_{i=0}^{m-2} k_i)-1}}, A_{1_{n-(\sum_{i=0}^{m-2} k_i)-2}}, \dots, A_{1_0})$$

\oplus

.

.

.

\oplus

$$(A_{m-1_{n-1}}, A_{m-1_{n-2}}, \dots, A_{m-1_{n-k_0}}, A_{m-1_{n-k_0-1}}, A_{m-1_{n-k_0-2}}, \dots, A_{m-1_{n-k_0-k_1}}, \dots, A_{m-1_{n-(\sum_{i=0}^{m-2} k_i)}},$$

$$a_{m-1_{n-(\sum_{i=0}^{m-2} k_i)-1}}, a_{m-1_{n-(\sum_{i=0}^{m-2} k_i)-2}}, \dots, a_{m-1_0})$$

互斥或是線性運算子，為了說明起見我們利用交換率將上式之會變動的

的位元均調整到最下方（此調整的動作只是為了證明無碰撞 (collision

free) 的特性，在實現時不必有此動作），原式成為：



五、發明說明 (11)

$$\begin{aligned}
 & (A_{m-1n-1}, A_{m-1n-2}, \dots, A_{m-1n-k_0}, A_{0n-k_0-1}, A_{0n-k_0-2}, \dots, A_{0n-(\sum_{i=0}^{m-2} k_i)}, A_{0n-(\sum_{i=0}^{m-2} k_i)-1}, A_{0n-(\sum_{i=0}^{m-2} k_i)-2}, \dots, A_{0_0}) \\
 & \oplus \\
 & (A_{1n-1}, A_{1n-2}, \dots, A_{1n-k_0}, A_{m-1n-k_0-1}, A_{m-1n-k_0-2}, \dots, A_{m-1n-(\sum_{i=0}^{m-2} k_i)}, A_{1n-(\sum_{i=0}^{m-2} k_i)-1}, A_{1n-(\sum_{i=0}^{m-2} k_i)-2}, \dots, A_{1_0}) \\
 & \oplus \\
 & \cdot \\
 & \cdot \\
 & \cdot \\
 & \oplus \\
 & (a_{0n-1}, a_{0n-2}, \dots, a_{0n-k_0}, a_{1n-k_0-1}, a_{1n-k_0-2}, \dots, a_{1n-k_0-k_1}, \dots, a_{m-2n-(\sum_{i=0}^{m-2} k_i)} \\
 & , a_{m-1n-(\sum_{i=0}^{m-2} k_i)-1}, a_{m-1n-(\sum_{i=0}^{m-2} k_i)-2}, \dots, a_{m-1_0})
 \end{aligned}$$

觀察上式，如果有二組輸入的值是不同的，表示至少有一會變動的位元不同，因為這些二組輸入不同值的位元均跟 $m-1$ 個固定值的位元作互斥或運算 (exclusive-OR) 運算，因此輸出的值也至少會有一個位元不同，我們可以斷言，只要輸入的值不同經過此索引法後也會產生不同的輸出值，因此可達到索引無碰撞 (collision free) 的效果。

多段編碼結構索引法可以應用到 VPI/VCI 這種兩段編碼結構的索引法中，由於非同步傳輸模式 (ATM) 交換機以設定 $\text{minimum}(VPI)$ 、 $\text{maximum}(VPI)$ 、 $\text{minimum}(VCI)$ 、 $\text{maximum}(VCI)$ 等參數來規範 VPI/VCI 的



五、發明說明 (12)

使用最為方便，我們特別再針對這種應用環境發展了一套方法也具有索引值無碰撞的效果，我們稱之為反向補償索引法 (Reverse Compensation Indexing Method)，說明如下：

若有二段編碼結構由 \bar{A}_i'' , $i=0,1$ 組成，每一編碼結構的值為

$\min(\bar{A}_i'') \leq \bar{A}_i'' \leq \max(\bar{A}_i'')$ ， k_i 滿足以下的數學式：

$$2^{k_i-1} < (\max(\bar{A}_i'') - \min(\bar{A}_i'') + 1) \leq 2^{k_i}$$

我們將每個 \bar{A}_i'' 分別加或減某值使轉換後的 \bar{A}_0' 低 k_0 個位元會變動， \bar{A}_1' 低 k_1 個位元會變動且 $\sum_{i=0}^1 k_i = n$ ，其中 n 是索引位元寬度。

\bar{A}_i' 均跟 $(2^n - 1)$ 做逐位元的 AND 運算，調整為 n 位元寬度的 \bar{A}_i' 。

$$\bar{A}_0' = (A_{0n-1}, A_{0n-2}, \dots, A_{0k_0}, a_{0k_0-1}, a_{0k_0-2}, \dots, a_{0_0})$$

$$\bar{A}_1' = (A_{1n-1}, A_{1n-2}, \dots, A_{1k_1}, a_{1k_1-1}, a_{1k_1-2}, \dots, a_{1_0})$$

我們將 \bar{A}_0' 之位元位置反向得 $\bar{A}_0 = (a_{0_0}, a_{0_1}, \dots, a_{k_0-1}, A_{0k_0}, A_{0k_0+1}, \dots, A_{0n-1})$ 再與

$\bar{A}_1 (= \bar{A}_1')$ 作互斥或運算得

$$(a_{0_0}, a_{0_1}, \dots, a_{0k_0-1}, A_{0k_0}, A_{0k_0+1}, \dots, A_{0n-1})$$

\oplus

$$(A_{1n-1}, A_{1n-2}, \dots, A_{1k_1}, a_{1k_1-1}, a_{1k_1-2}, \dots, a_{1_0})$$



五、發明說明 (13)

互斥或是線性運算子符合交換率，用交換率調整上式(此調整的動作只是為了證明無碰撞(collision free)的特性，在實現時不必有此動作)，原式成為：

$$(A_{i_{n-1}}, A_{i_{n-2}}, \dots, A_{i_{k_1}}, A_{0_{k_0}}, A_{0_{k_0+1}}, \dots, A_{0_{n-1}})$$

\oplus

$$(a_{0_{n-1}}, a_{0_{n-2}}, \dots, a_{0_{k_0-1}}, a_{i_{k_1-1}}, a_{i_{k_1-2}}, \dots, a_{i_0})$$

觀察上式，因為 $\sum_{i=0}^1 k_i = n$ ，如果有二組輸入的值不同，表示某些位元的值不同，因此輸出的值在對應的位元位置一定會不一樣，我們可以斷言，只要輸入的值不同經過此索引法後也會產生不同的輸出值，因此可達到索引無碰撞(collision free)的效果。由於以上 \bar{A}_0 需做位元反向的動作，因此本方法稱為反向補償索引法。

以下舉幾個實施例說明本索引法如何應用於現行非同步傳輸模式(ATM)網路。



五、發明說明 (14)

實施例一：

VPI/VCI 值為 \bar{A}_i^m , $i=0,1$ 型式時，即運作在任意 $\min(VPI)$, $\max(VPI)$, $\min(VCI)$, $\max(VCI)$ 的場合中，我們假設 $\min(VPI)=7$, $\max(VPI)=8$, $\min(VCI)=1$, $\max(VCI)=8$ 來說明。

步驟 1：

用以下的規範找出 k_0, k_1, n 的值：

$$2^{k_0-1} < (\max(VPI) - \min(VPI) + 1) \leq 2^{k_0}$$

$$\Rightarrow 2^{k_0-1} < (8-7)+1 \leq 2^{k_0}$$

$$\Rightarrow 2^{k_0-1} < 2 \leq 2^{k_0}$$

$$\Rightarrow k_0 = 1$$

$$2^{k_1-1} < (\max(VCI) - \min(VCI) + 1) \leq 2^{k_1}$$

$$\Rightarrow 2^{k_1-1} < (8-1)+1 \leq 2^{k_1}$$

$$\Rightarrow 2^{k_1-1} < 8 \leq 2^{k_1}$$

$$\Rightarrow k_1 = 3$$

$$\text{索引位元寬度 } n = \sum_{i=0}^1 k_i = k_0 + k_1 = 1 + 3 = 4$$



五、發明說明 (15)

步驟 2：

令 $\bar{A}_0'' = \bar{A}_0''' + 1$, $\bar{A}_1' = \bar{A}_1''' - 1$, (如多段編碼結構索引法敘述中所舉之特例 ,

亦可使 $\bar{A}_0' = \bar{A}_0''' - \min(VPI)$, $\bar{A}_1' = \bar{A}_1''' - \min(VCI)$) 。

$8 \leq \bar{A}_0' \leq 9 \Rightarrow (1000)_2 \leq \bar{A}_0' \leq (1001)_2$ 可知 \bar{A}_0' 只有右邊一個位元(即右邊 k_0 個位元)會變動。

$0 \leq \bar{A}_1' \leq 7 \Rightarrow (000)_2 \leq \bar{A}_1' \leq (111)_2$ 可知 \bar{A}_1' 只有右邊三個位元(即右邊 k_1 個位元)會變動。

步驟 3：

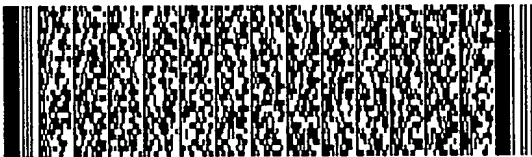
\bar{A}_0' , \bar{A}_1' 均跟 $(2^n - 1) = (2^4 - 1) = 15 = (1111)_2$ 做逐位元的 AND 運算 , 調整為 4 位元寬度的 \bar{A}_0' , \bar{A}_1' 。

$$\bar{A}_0' = (1, 0, 0, a_{0_0})_2 \Rightarrow (1, 0, 0, 0)_2 \leq \bar{A}_0' \leq (1, 0, 0, 1)_2$$

$$\bar{A}_1' = (0, a_{1_2}, a_{1_1}, a_{1_0})_2 \Rightarrow (0, 0, 0, 0)_2 \leq \bar{A}_1' \leq (0, 1, 1, 1)_2$$

步驟 4：

我們將 \bar{A}_0' 之位元位置反向得 $\bar{A}_0 = (a_{0_0}, 0, 0, 1)_2$ 再與 $\bar{A}_1 (= \bar{A}_1')$ 作互斥或運算得



五、發明說明 (16)

$$(a_{0_0}, 0, 0, 1)_2$$

\oplus

$$(0, a_{1_2}, a_{1_1}, a_{1_0})_2$$

用交換率調整上式得：

$$(0, 0, 0, 1)_2$$

\oplus

$$(a_{0_0}, a_{1_2}, a_{1_1}, a_{1_0})_2$$

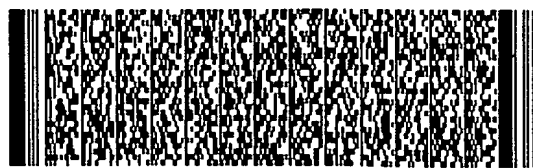
觀察上式可知可達到索引值無碰撞的效果，如圖 2 所示為實施例一的參考電路。若 n 位元減法器的延遲為 t_s ，互斥或閘的延遲為 t_x （單位均為秒），則此索引電路配合整個索引法可支援每秒鐘 $1/(t_i + t_a + t_m) = 1/(t_s + t_x + t_a + t_m)$ 個 ATM 碼格(Cell)的速率，其中 t_i 是索引值運算時間， t_a 是位址產生時間， t_m 是記憶體存取時間。

實施例二：

VPI/VCI 值為 \bar{A}_i ， $i=0,1$ 型式時，即運作在 \bar{A}_i 低 k_i 個位元會變動且

$$\sum_{i=0}^1 k_i = n，如 \min(VPI)=8, \max(VPI)=9, \min(VCI)=0, \max(VCI)=7。$$

索引法仿實施例一，但跳過步驟二的將 \bar{A}_0 ， \bar{A}_1 分別加或減某值的程序，



五、發明說明 (17)

因為原本的 \bar{A}_0 低 k_0 個位元會變動， \bar{A}_1 低 k_1 個位元會變動；詳細步驟不再贅述。

實施例三：

本實施例為實施例二的特例，說明反向補償索引法在某些場合之卓越特性：

問題描述：一般說來， VPI 及 VCI 會由零開始使用，因此，我們來討論是否有一種電路可完全因應以下的場合：

$\min(VPI)=0$ ， $\max(VPI)=(2^{k_0}-1)$ ， $\min(VCI)=0$ ， $\max(VCI)=(2^{k_1}-1)$ ，其中 $\sum_{i=0}^1 k_i = n$ ， n 為索引位元寬度。

我們可將 \bar{A}_0 (即 VPI) 之位元位置反向得

$\bar{A}_0 = (a_{0_0}, a_{0_1}, \dots, a_{0_{k_0-1}}, 0, 0, \dots, 0)$ 再與 $\bar{A}_1 (= \bar{A}_1')$ (即 VCI) 作互斥或運算得

$(a_{0_0}, a_{0_1}, \dots, a_{0_{k_0-1}}, 0, 0, \dots, 0)$

\oplus

$(0, 0, \dots, 0, a_{1_{k_1-1}}, a_{1_{k_1-2}}, \dots, a_{1_0})$

觀察上式，因為會變動的位元均跟 0 作互斥或運算 (exclusive-OR) 運算，可知可達到索引值無碰撞的效果。

五、發明說明 (18)

換句話說，這個實施例說明使用反向補償索引法在實施例三的場合可具的卓越特性為只要 $\min(VPI)=0$ 及 $\min(VCI)=0$ ，用同一套電路可適用於

$\max(VPI) = 2^{k_0} - 1$
 $\max(VCI) = 2^{n-k_0} - 1$ ， $k_0 = 0, 1, \dots, n$ 等 $(n+1)$ 種場合都具有索引值無碰撞的效果。

這樣的電路我們在圖 3 中表示，在此舉個簡單的實施例若 $n=8$ ，圖 3 的電路使用 8 個二輸入的互斥或閘，其可提供以下 9 種場合都具有索引值無碰撞的效果：

第一種場合 $\max(VPI)=0$ ， $\max(VCI)=255$ 。

第二種場合 $\max(VPI)=1$ ， $\max(VCI)=127$ 。

第三種場合 $\max(VPI)=3$ ， $\max(VCI)=63$ 。

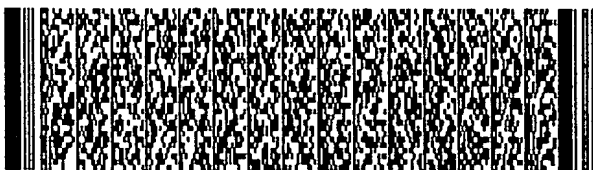
第四種場合 $\max(VPI)=7$ ， $\max(VCI)=31$ 。

第五種場合 $\max(VPI)=15$ ， $\max(VCI)=15$ 。

第六種場合 $\max(VPI)=31$ ， $\max(VCI)=7$ 。

第七種場合 $\max(VPI)=63$ ， $\max(VCI)=3$ 。

第八種場合 $\max(VPI)=127$ ， $\max(VCI)=1$ 。



五、發明說明 (19)

第九種場合 $\max(VPI)=255$ ， $\max(VCI)=0$ 。

如圖 3 為適用於實施例二、實施例三場合的參考電路，可在一個互斥或閘的延遲時間完成索引值運算。反向補償索引法的電路運作於整個索引法中可支援到每秒 $1/(t_i+t_o+t_m)$ 個非同步傳輸模式(ATM)碼格(Cell)的速率，若互斥或閘的延遲為 t_x (單位為秒)，則此時之 $t_i=t_x$ ，其中 t_i 是索引值運算時間， t_o 是位址產生時間， t_m 是記憶體存取時間。

非同步傳輸模式(ATM)接收單元若為了在支援不同 VPI/VCI 連線數時，將安裝在接收單元的索引表格記憶體相對地增減，可配合如圖 4 的表格位元寬度調整器 (TBWA, Table Bit Width Adjuster) 電路，其具有可根據索引表格記憶體的位元寬度來改變 VPI 及 VCI 為輸入的互斥或運算(exclusive-OR)組態的功能；舉例來說，若 $TBW=3$ ，則 XOR 組態為 $VPI[0]$ 要跟 $VCI[2]$ ， $VPI[1]$ 要跟 $VCI[1]$ ， $VPI[2]$ 要跟 $VCI[0]$ 分別作互斥或運算。若 $TBW=4$ ，則互斥或運算 (exclusive-OR) 組態變為 $VPI[0]$ 要跟 $VCI[3]$ ， $VPI[1]$ 要跟 $VCI[2]$ ， $VPI[2]$ 要跟 $VCI[1]$ ， $VPI[3]$ 要跟 $VCI[0]$ 分別作互斥或運算。



五、發明說明 (20)

接下來探討此索引法如何應用在 ATM 交換機的接收埠，以快速地找到下一個階段的輸出埠以及新的 VPI/VCI 值。非同步傳輸模式(ATM)交換機具有兩個或兩個以上的接收埠，我們也可將如圖 4 的電路直接應用在每一個接收埠上，如圖 5 所示為有 r 個(編號 0 到 $r-1$)接收埠的交換機的索引電路，每一個接收埠之索引表格的基礎位址分別是 $BP_i, i=0,1,\dots,r-1$ ，(BP 是 Base Pointer 的縮寫)，但由圖 5 的架構圖可看出因為個別使用自己的索引表格記憶體，系統建置好後如果某個接收埠需要超過原本的索引表格記憶體支援的連線數，就算其他接收埠有剩餘的索引表格記憶體，也無法機動調配給此接收埠運用，為了解決此種問題我們提出如圖 6 的架構圖。

如圖 6 右端為具有 e 個資料條目(Entry 0, Entry 1, ..., Entry $e-1$)的索引表格記憶體， $\sum_{i=0}^{r-1} 2^{TBW_i} = e$ ，第 0 個接收單元的基礎位址是 BP_0 ，其他 $r-1$ 個接收埠的基礎位址(Base Pointer)均以串接方式從第 0 個接收單元的表格位元寬度(TBW)及基礎位址(BP)計算出下一個接收埠的基礎位址，其計算式為 $BP_i = BP_{i-1} + W * 2^{TBW_{i-1}}, i=1,\dots,r-1$ ，算出每一個位址(BP)的目的是使得不同的接收埠使用的索引表格記憶體的區間得以依序分



五、發明說明 (21)

配而不衝突，這種共享記憶體(Shared Memory)的架構可在一定大小的共享記憶體的情況中更有效地調整每一個接收埠支援的連線數。

經過以上的討論後，我們可很容易地推演出如何將反向補償索引法應用於多段編碼結構中，使得多段編碼結構索引法更具使用上的彈性；其方法為將多段編碼結構依需要兩兩分組，每一組的兩段結構均可視為實施反向補償索引法的基本單元，因此多段編碼結構也可呈現出反向補償索引法的使用彈性。

綜上所述，本發明具備原創性、新穎性及進步性。雖然本發明以一些較佳實施例揭露如上，然其並非用以限定本發明，任何熟習此技術者，在不脫離本發明之精神和範圍內，當可作為些許之更動與潤飾，根據此發明之精神，設計出對應的多段編碼結構索引法及反向補償索引法之線路，因此本發明之保護範圍當視後附之申請專利範圍所界定為準。



圖式簡單說明

圖式說明：

圖 1 在非同步傳輸模式(ATM)接收端的 VPI/VCI 索引法的示意圖

圖 2 為例一的參考電路

圖 3 使用 8 個二輸入的互斥或閘的電路

圖 4 表格位元寬度調整器(TBWA)電路

圖 5 有 r 個(編號 0 到 $r-1$)接收埠的交換機的索引電路

圖 6 架構圖

圖號說明：

101.... 輸入 (Input)

102.... 索引函數 (Indexing function)

103.... n 位元索引值

104.....基礎位址 (Base pointer)

201.... n 位元減法器

202.... 被減數

203.... 減數

204.... 差



圖式簡單說明

401.... 表格位元寬度(TBW)位元的索引值

402.... 表格位元寬度 (TBW, Table Bit Width)



六、申請專利範圍

1. 一種多段編碼結構索引法的實施方法，其係包括

步驟 1： m 段編碼結構由 \bar{A}_i^m ， $i=0,1,\dots,m-1$ 組成， \bar{A}_i^m 最多有 2^{k_i} 種變化；

步驟 2：我們可利用某種轉換方法為將原本的 \bar{A}_i^m 各個轉換為 \bar{A}_i^n ，且 \bar{A}_i^n 有 k_i 個位元會變動， $\sum_{i=0}^{m-1} k_i = n$ ； n 為本索引法的索引位元寬度；

步驟 3： \bar{A}_i^n 均跟 $(2^n - 1)$ 做逐位元的 AND 運算，可調整為 n 位元寬度的 \bar{A}_i^n ；

步驟 4：將 \bar{A}_i^n 調整成如下的型式作 $\bar{A}_0 \oplus \bar{A}_1 \oplus \dots \oplus \bar{A}_{m-1}$ 運算； \bar{A}_i^n 的第 $n - (\sum_{j=0}^{i-1} k_j + 1)$ 到第 $n - \sum_{j=0}^i k_j$ 位元均為會變動的位元；（在此為使表示式簡潔起見，令 $\sum_{j=0}^{-1} k_j = 0$ ）； $\bar{A}_0 \oplus \bar{A}_1 \oplus \dots \oplus \bar{A}_{m-1}$ 其有只要輸入的值不同，經過此索引法後也會產生不同的輸出值的特性，也就是說可達到索引無碰撞(collision free)的效果。

2. 如申請專利範圍第 1 項的的實施方法，其步驟 2 中提出兩種可依需要個別或混合使用的方法，這兩個方法的目的均是使得每一 k_i 能

夠儘量小，以期減少索引位元度 n ，其中 $\sum_{i=0}^{m-1} k_i = n$ 。



六、申請專利範圍

3. 如申請專利範圍第 2 項的的實施方法，其第一種轉換的方法是觀察原本的每一個 \bar{A}_i'' ，若會變動的位元並不是在最右側，則可將會變動的位元調整到最右側。

4. 如申請專利範圍第 2 項的的實施方法，其第二種轉換方法可適用於以下的條件下：若其中某些編碼結構的值為

$\min(\bar{A}_i'') \leq \bar{A}_i'' \leq \max(\bar{A}_i'')$ ， k_i 滿足以下的數學式：

$$2^{k_i-1} < (\max(\bar{A}_i'') - \min(\bar{A}_i'') + 1) \leq 2^{k_i}$$

我們可將這些編碼結構 \bar{A}_i'' 分別加或減某值使轉換後的 \bar{A}_i' 低 k_i 個位元會變動。可以減的值的特例剛好是 $\min(\bar{A}_i'')$ 。

5. 一種反向補償索引法的實施方法，其係包括：

步驟 1：二段編碼結構由 \bar{A}_i'' ， $i=0,1$ 組成，每一編碼結構的值為

$\min(\bar{A}_i'') \leq \bar{A}_i'' \leq \max(\bar{A}_i'')$ ， k_i 滿足以下的數學式：

$$2^{k_i-1} < (\max(\bar{A}_i'') - \min(\bar{A}_i'') + 1) \leq 2^{k_i}$$

步驟 2：將每個 \bar{A}_i'' 分別加或減某值使轉換後的 \bar{A}_i' 低 k_i 個位元會

變動， \bar{A}_1' 低 k_1 個位元會變動且 $\sum_{i=0}^1 k_i = n$ ，其中 n 是索引位元寬度；



六、申請專利範圍

步驟 3： \bar{A}_i 均跟 $(2^n - 1)$ 做逐位元的 AND 運算，調整為 n 位元寬度的 \bar{A}_i ；

步驟 4：我們將 \bar{A}_0 之位元位置反向得到之 \bar{A}_0 再與 $\bar{A}_1 (= \bar{A}_1)$ 作互斥或運算，即 $\bar{A}_0 \oplus \bar{A}_1$ ；其有只要輸入的值不同，經過此索引法後也會產生不同的輸出值的特性，也就是說可達到索引無碰撞(collision free)的效果。

6. 如申請專利範圍第 5 項的的實施方法，其反向補償索引法的卓越特性為只要 $\min(VPI)=0$ 及 $\min(VCI)=0$ ，用 n 個互斥或閘的電路可適用於：

$\max(VPI) = 2^{k_0} - 1$
 $\max(VCI) = 2^{n-k_0} - 1$ ， $k_0 = 0, 1, \dots, n$ 等 $(n+1)$ 種場合都具有索引值無碰撞的效果。

7. 一種運用多段編碼結構對應到單一表格的方法，其係 ATM 接收單元在支援不同的 VPI/VCI 連線數時，將安裝在接收單元的索引表格記憶體相對地增減，可配合 TBWA 電路，其具有可根據索引表格記憶體的位元寬度來改變 VPI 及 VCI 為輸入的互斥或運算



六、申請專利範圍

(exclusive-OR)組態的功能。

8. 一種運用多段編碼結構對應到共享記憶體(Shared Memory)的架構

，其係將具有 e 個資料條目(Entry 0, Entry 1, ., Entry $e-1$) 的索引表格記憶體， $\sum_{i=0}^{r-1} 2^{TBW_i} = e$ ，第 0 個接收單元的基礎位址是 BP_0 ，其他 $r-1$ 個接收埠的基礎位址 (Base Pointer) 均以串接方式從第 0 個接收單元的表格位元寬度 (TBW) 及基礎位址 (BP) 計算出下一個接收埠的基礎位址，其計算式為 $BP_i = BP_{i-1} + W * 2^{TBW_{i-1}}, i = 1, \dots, r-1$ ，算出每一個 BP 的目的是使得不同的接收埠使用的索引表格記憶體的區間得以依序分配而不衝突，這種共享記憶體(Shared Memory)的架構可在一定大小的共享記憶體的情況中更有效地調整每一個接收埠支援的連線數。

9. 一種運用多段編碼結構的方法，其係依需要將多段編碼結構兩兩分組，每一組的兩段結構均可視為實施反向補償索引法的基本單元，因此多段編碼結構也可呈現出反向補償索引法的使用彈性。



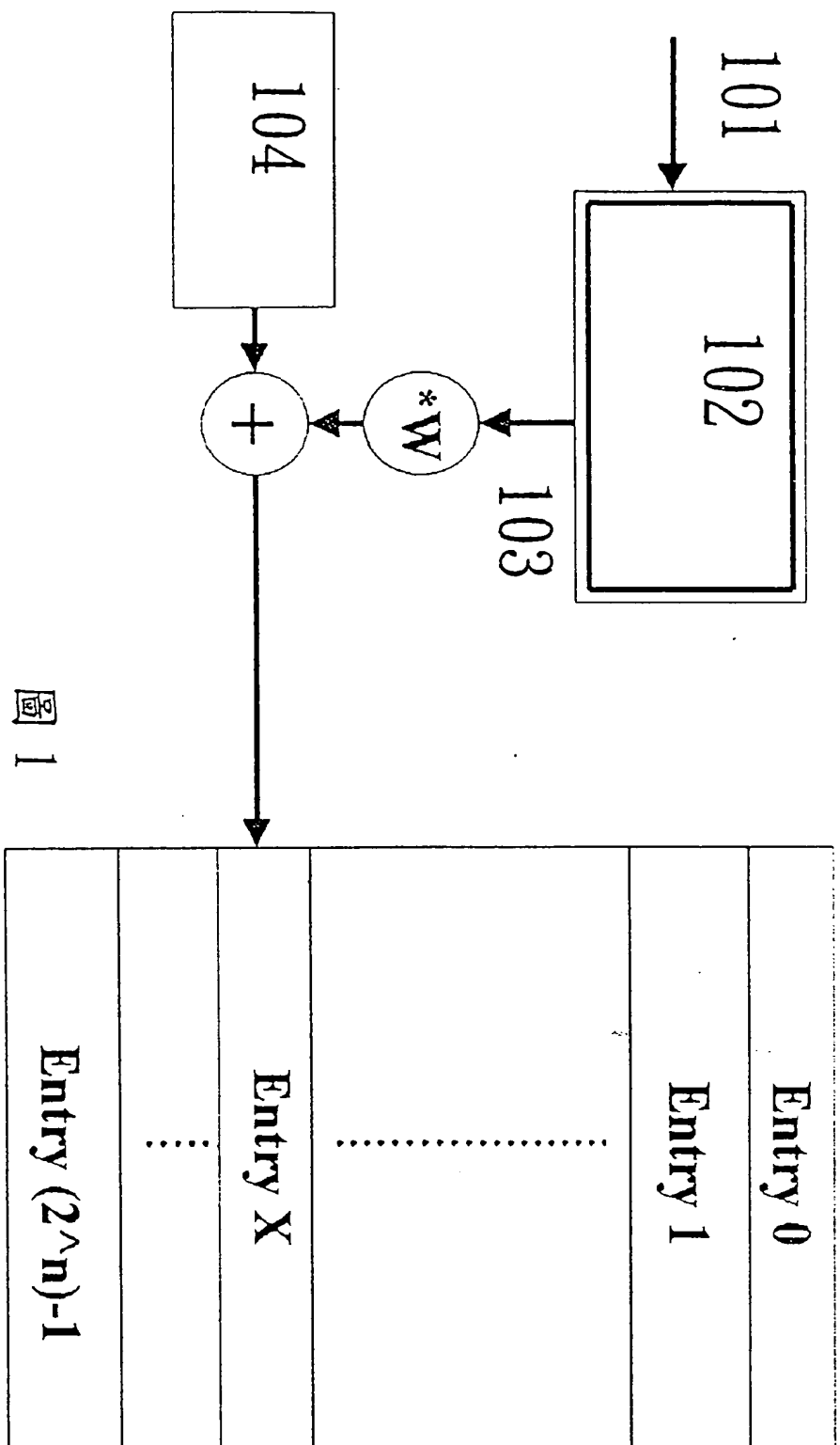


圖 1

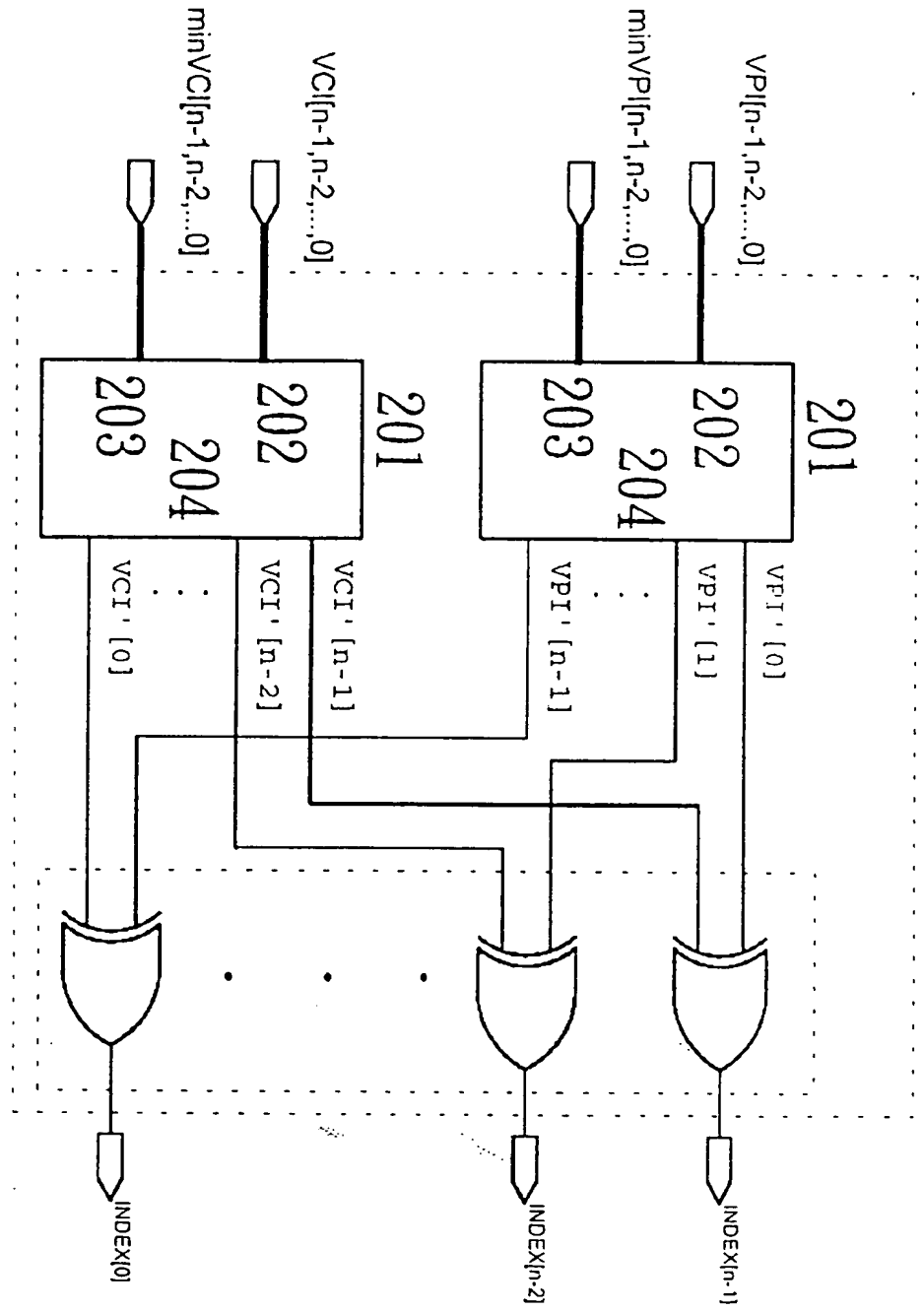


圖 2

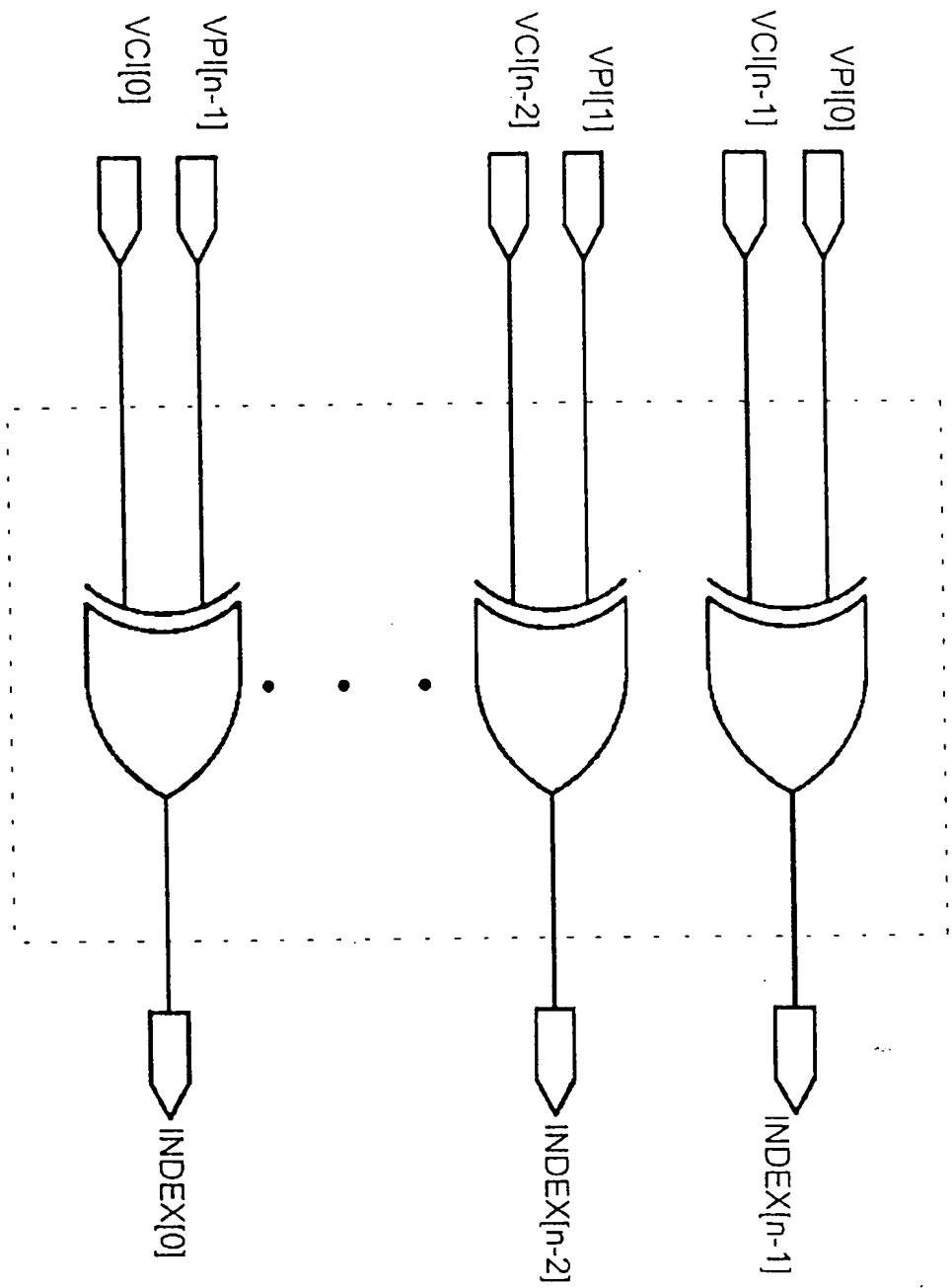
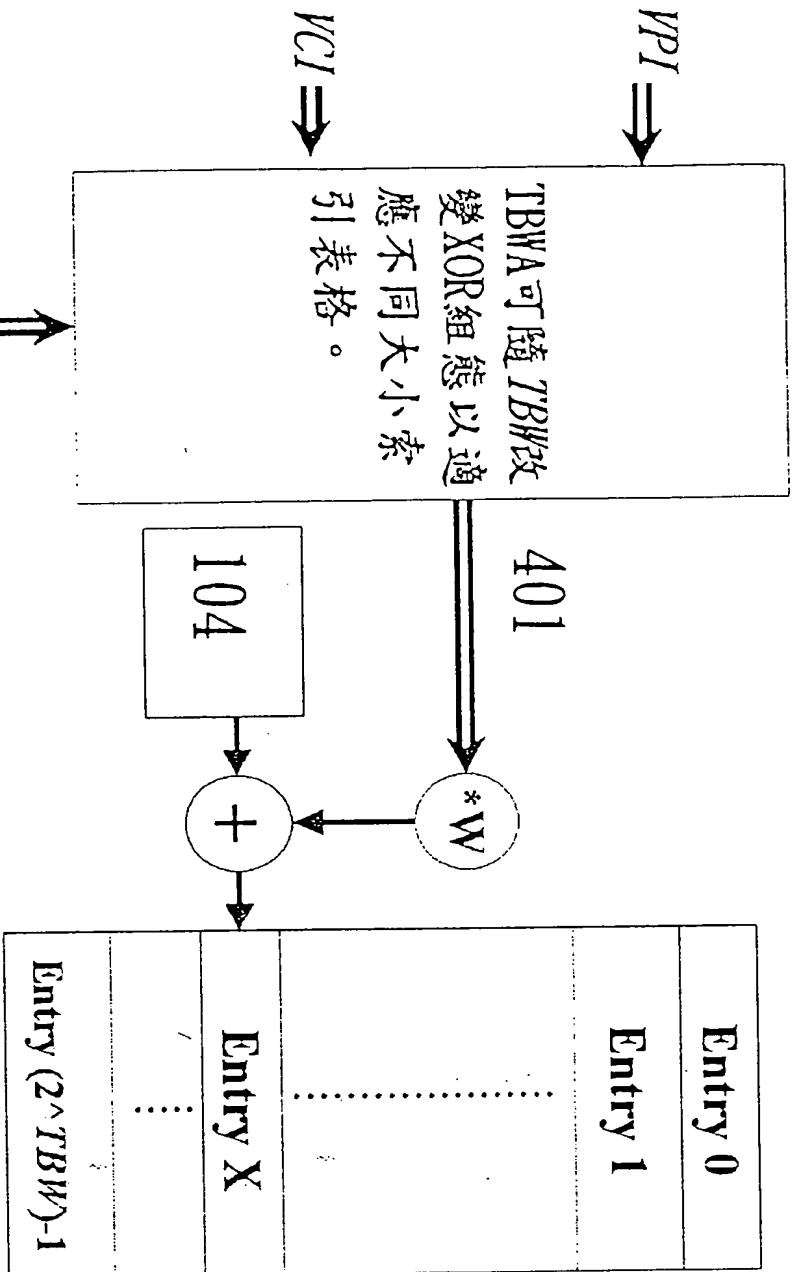


圖 3



402

圖 4

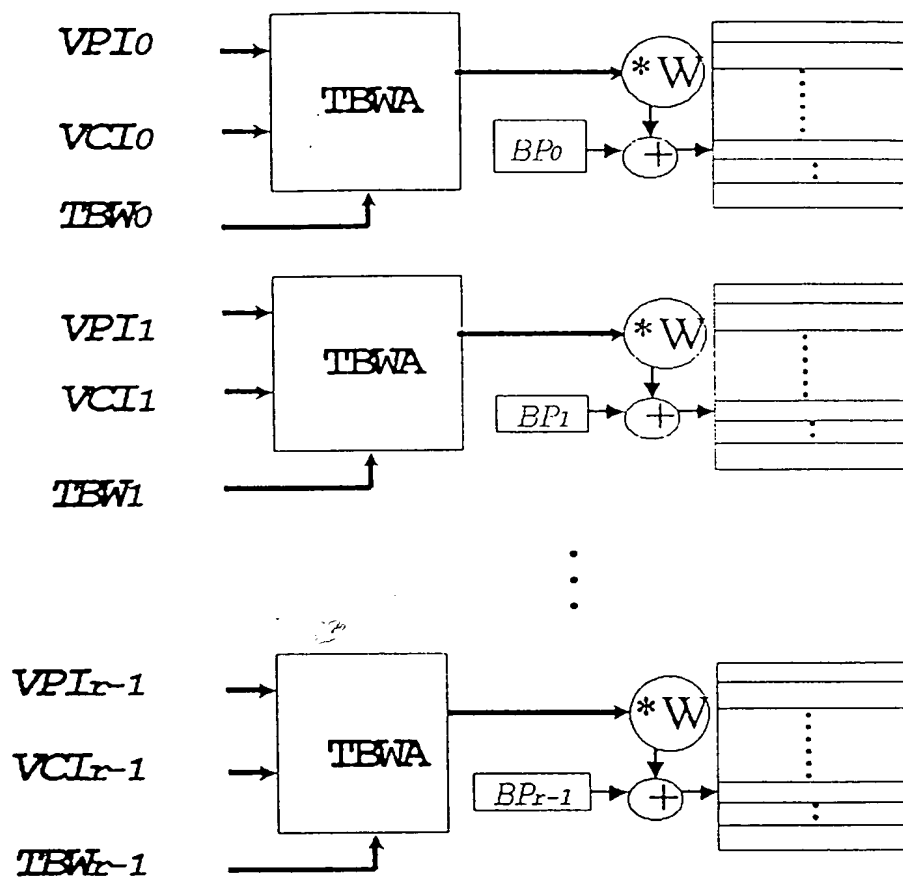
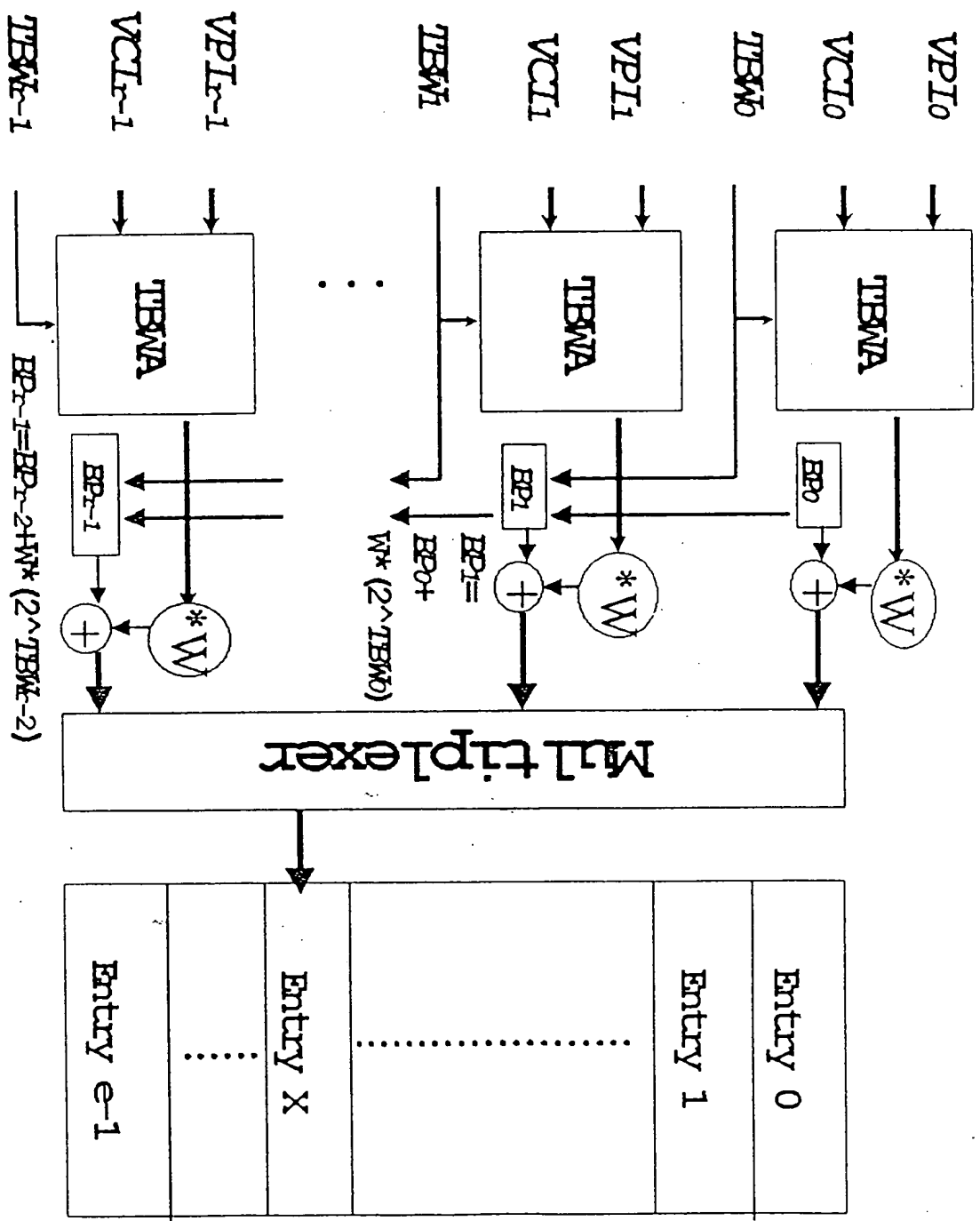


圖 5



$$2^{TBM_0} + 2^{TBM_1} + \dots + 2^{TBM_{i-1}} = 0$$

圖 6